

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZAČNÍ A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF AUTOMATISATION AND MEASUREMENT

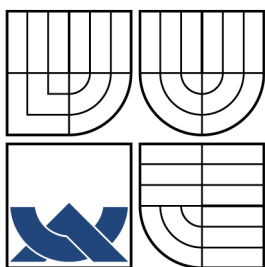
MULTIFUNKČNÍ VÝŠKOMĚR

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

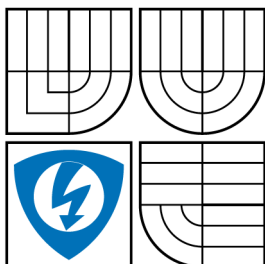
AUTOR PRÁCE
AUTHOR

MIROSLAV LÍZNER

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZAČNÍ A MĚŘICÍ TECHNIKY



FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF AUTOMATISATION AND
MEASUREMENT

MULTIFUNKČNÍ VÝŠKOMĚR MULTIFUNCTIONAL ALTIMETER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MIROSLAV LÍZNER

VEDOUcí PRÁCE
SUPERVISOR

ING. STANISLAV KLUSÁČEK

BRNO 2009

ABSTRAKT

Tato bakalářská práce se zabývá připojením snímačů tlaku, vlhkosti a teploty k mikrokontroléru. Cílem je navrhnout turistický výškoměr umožňující záznam měřených hodnot. Praktická část popisuje konkrétní realizaci založenou na mikrokontroleru ATmega64 a návrh programu v jazyce C.

KLÍČOVÁ SLOVA

Výškoměr, nadmořská výška, tlak, teplota, vlhkost.

ABSTRACT

This bacheleor thesishis deals with pressure, humidity and teperature sensors to microcontroller connection possibilities. The main goal of this work is to develop the recording touristic altimeter. The hardware solution based on ATmega64 microcontroller and software implementation is described in practical part of this work.

KEYWORDS

Altimeter, altitude, pressure, temperature, humidity

LÍZNER M. *Multifunkční výškoměr*. Brno: Vysoké učení technické. Fakulta elektrotechniky a komunikačních technologií. Ústav automatizace a měřicí techniky, 2009. 51 stran, 8 stran příloh. Bakalářská práce. Vedoucí práce byl Ing. Stanislav Klusáček.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma Multifunkční výškoměr jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

Děkuji vedoucímu bakalářské práce Ing. Stanislavu Klusáčkovi za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

OBSAH

Úvod	11
1 Měření meteorologických veličin	12
1.1 Atmosferický tlak	12
1.2 Teplota vzduchu	13
1.3 Vlhkost vzduchu	13
1.4 Rosný bod	13
1.5 Měření výšky	14
2 Senzory	16
2.1 Senzor absolutního tlaku Motorola MPX4115	16
2.2 Snímač vlhkosti Sensirion SHT11	17
2.2.1 Přenos dat	18
3 Použitá rozhraní	21
3.1 SPI	21
3.1.1 Zapojení SPI	21
3.1.2 Technické parametry	21
3.2 USB	22
3.2.1 Zapojení USB	23
3.2.2 Základní vlastnosti	23
4 Návrh hardwarové části	24
4.1 Procesor Atmel ATmega64a	24
4.2 Snímací obvody	30
4.2.1 Měření tlaku	30
4.2.2 Měření vlhkosti	31
4.3 A/D převodník MCP3202	31
4.4 Napájení	32
4.5 Displej	33
4.6 Komunikace po USB	35
4.7 Mechanické provedení	35
5 Návrh softwarové části	37
5.1 Konstanty a makra	37
5.2 Komunikace s A/D převodníkem	38
5.3 Výpočet hodnot tlaku, teploty a nadmořské výšky	39

5.4	Výpis na displej	40
5.5	Sériový přenos dat	42
5.6	Ukládání dat	43
6	Výsledky práce	45
6.1	Osazení a realizace hardwarové části	45
6.2	Software a test funkčnosti	45
6.3	Parametry přístroje	46
7	Závěr	47
	Reference	48
	Seznam symbolů, veličin a zkratk	50
	Seznam příloh	51
A	Výkresová dokumentace	52
A.1	Schéma zapojení	52
A.2	Tištěný spoj - strana spojů	53
A.3	Tištěný spoj - strana součástek	53
A.4	Osazovací plánec - vrchní strana	54
A.5	Osazovací plánec - spodní strana	54
A.6	Tištěný spoj - deska s tlačítky	55
A.7	Přístrojová krabička	56
B	Seznam součástek	57
C	Software - vývojový diagram	59

SEZNAM OBRÁZKŮ

1.1	Synoptická mapa [13]	12
2.1	Obrázek snímače MPX4115 [4]	16
2.2	Závislost výstupního napětí snímače MPX4115 na absolutním tlaku [4]	17
2.3	Ukázka měřící sekvence SHT11	18
2.4	Startovací sekvence "Transmission start"	18
3.1	Schema zapojení SPI pro dva mikroprocesory	22
3.2	USB konektory	23
4.1	Blokové schéma zapojení	24
4.2	Blokový diagram AVR architektury [3]	26
4.3	Signály na ISP konektoru výškoměru	29
4.4	USART - Formát rámce	30
4.5	Zapojení napájecího obvodu	33
4.6	Schéma zapojení displeje	33
4.7	Schéma zapojení desky s tlačítky	34
4.8	Zapojení obvodu FT232R	36
4.9	Neosazená deska tištěného spoje	36
5.1	SPI komunikace po osmibitových segmentech [11]	39
5.2	Zobrazení aktuálních hodnot na displeji	41
6.1	Výsledné mechanické provedení zařízení	45

SEZNAM TABULEK

2.1	Základní parametry snímače MPX4115	17
2.2	Příkazy pro komunikaci se snímačem SHT11	18
2.3	Status registr obvodu SHT11	19
2.4	Koeficienty pro linearizaci čidla SHT11	19
2.5	Technické parametry snímače Sensirion SHT11 pro měření vlhkosti .	19
2.6	Technické parametry snímače Sensirion SHT11 pro měření teploty . .	20
3.1	Zapojení USB	23
4.1	Fuse Low	27
4.2	Nastavení hodinového signálu	28
4.3	Fuse High	28
4.4	Extended Fuse	28
4.5	Lock Bits	29
4.6	Význam konfiguračních bitů MCP3202	32
4.7	Zapojení displeje DIP128-6	34
6.1	Parametry přístroje	46

ÚVOD

Cílem bakalářské práce je navrhnout hardware a software pro multifunkční výškoměr. V následujícím textu bude popsán návrh přístroje, který bude schopen měřit atmosferický tlak, nadmořskou výšku, relativní ahodnotu vlhkosti vzduchu, rosný bod a teplotu okolního prostředí. Protože je přístroj navrhován s ohledem na využití v turistice, dokáže zaznamenávat měřené hodnoty které lze následně stáhnout do počítače. Lze tak například zaznamenat členitost terénu během výstupu v horách a po skončení výletu si výsledky zobrazit v počítači.

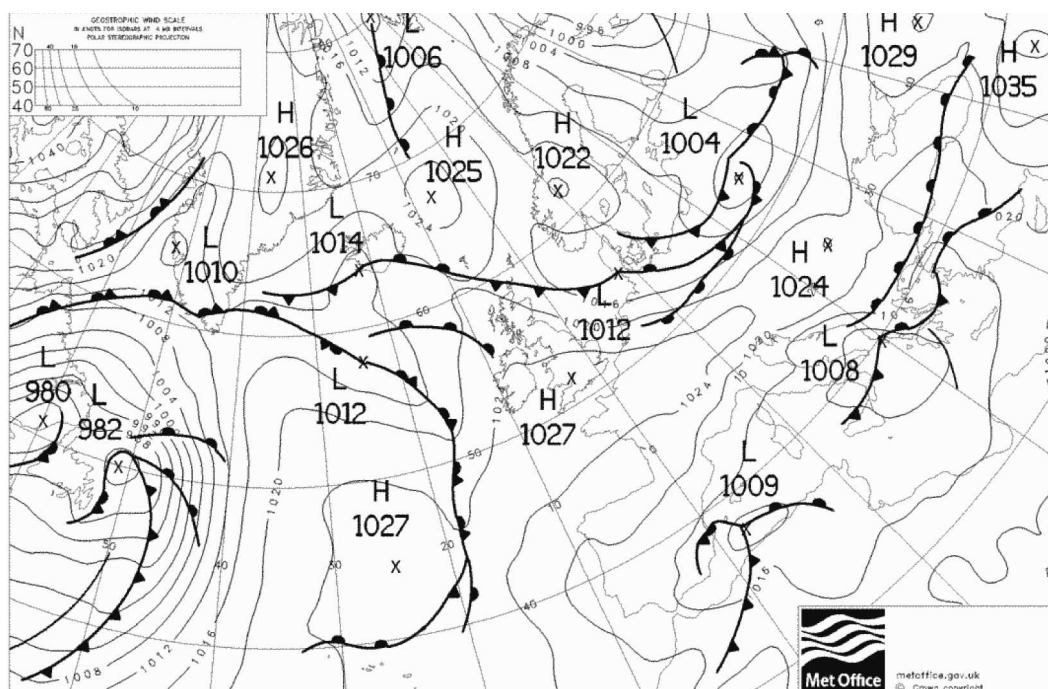
Práce je rozdělena do několika částí. V první části zmíním základní informace o měření meteorologických veličin. Další kapitola popisuje možnosti a parametry vybraných snímačů. Následují informace o použitých rozhraních (SPI, USB). Další kapitola se zabývá vlastním návrhem hardwarové části přístroje. V této části se podrobněji věnuji hlavním částem obvodu a výsledné mechanické konstrukci. V poslední části se již zabývám návrhem vhodných funkcí pro ovládání jednotlivých částí zapojení a výslednou koncepcí řídicího programu.

1 MĚŘENÍ METEOROLOGICKÝCH VELIČIN

1.1 Atmosferický tlak

Hmotnost vzduchu působí na zemský povrch tlakem (hydrostatický tlak). Velikost tlaku závisí na množství vzduchu ležícího nad danou plochou. Z toho důvodu je nejvyšší tlak u povrchu země a s rostoucí výškou klesá. Atmosferický tlak se obvykle měří v hektopascalech (hPa). Jedná se o značně proměnlivou veličinu, jelikož kolísání probíhá bez jakékoliv pravidelnosti. Důvody změn tlaku vzduchu může způsobovat nepravidelné ohřívání zemského povrchu, výměna teplého vzduchu za studený, nebo nahromadění vzduchu v jedné oblasti a naopak odčerpání z jiné. Měření tlaku je velmi podstatné pro předpověď počasí. Zvýšení tlaku signalizuje obvykle slunečné počasí, pokles spíše oblačno a deštivé počasí.

Rozdělení tlaku se na synoptických mapách znázorňuje pomocí izobar. Izobara je čára spojující na mapě místa se stejnou hodnotou tlaku vzduchu, přepočítanou na hladinu moře. Přístroj, který zaznamenává tlak se nazývá barometr.



Obrázek 1.1: Synoptická mapa [13]

Průměrný atmosférický tlak u hladiny moře se nazývá normální atmosférický tlak a má hodnotu 1013,25 hPa. Při výstupu o každých 5,5km se tlak snižuje na polovinu. Závislost tlaku na výšce vyjadřuje tzv. barický stupeň, jehož hodnota závisí na výšce

a teplotě a činí přibližně 8m na 1hPa ve výškách okolo 1000mm, 15m/hPa v 5000 mm a 30m/hPa v 10000mm. Pro běžnou potřebu v našich nadmořských výškách lze říci, že absolutní tlak (tj. tlak v místě měření) lze přepočítat na tlak na hladině moře tak, že na každých vystoupaných 8m nadmořské výšky odečteme 1 hPa.

1.2 Teplota vzduchu

Vzduch se stejně jako každá jiná hmota či těleso vyznačuje jistou teplotou. Ta je dána termodynamickou veličinou charakterizující kinetický stav molekul a atomů.

V meteorologii se teplota vzduchu měří ve výšce 2m od země v meteorologické budce, naměřené hodnoty se udávají podle Celsiovy stupnice. Měření teploty vychází podle mezinárodních dohod ze dvou pevných bodů - bodu mrazu (označenou v Celsiově stupnici jako 0 °C) a bodu varu (100 °C). Teplota je v různých výškách různá - v nižších polohách je teplota vyšší a ve vyšších polohách nižší. Může však nastat i stav, kdy tomu tak nebude. Tento zvrát teploty vzduchu se nazývá inverze.

1.3 Vlhkost vzduchu

Vlhkost vzduchu je množství vodních par obsažených ve vzduchu. Vodní páry obsažené ve vzduchu jsou podmínkou pro vznik oblačnosti a srážek. Mírou nasycení vzduchu vodní parou je relativní vlhkost. Je to poměr mezi skutečným obsahem vodních par a maximálním možným obsahem par při dané teplotě. Relativní vlhkost je udávána v procentech. 100% relativní vlhkosti znamená nasycení vzduchu vodní parou. Tento případ nastává např. při mlhách. Teplota, při níž se pára obsažená ve vzduchu stane nasycenou, se nazývá rosný bod. Průměrná relativní vlhkost je nejvyšší v listopadu, prosinci a lednu. Nejnižší je mezi dubnem až srpnem.

1.4 Rosný bod

V předchozím odstavci jsem zmínil rosný bod jako teplotu při 100% relativní vlhkosti. Jedná se o teplotu, při které je vzduch nasycen vodními parami bez dodání vodní páry z vnějšku. Při poklesu teploty prostředí pod tuto teplotu obvykle dochází ke kondenzaci vodní páry ve vzduchu a vzniká tak mlha či rosa. Pokud je relativní vlhkost vzduchu nižší než 100%, je teplota rosného bodu vždy nižší než teplota vzduchu. Rosný bod se využívá v mnoha meteorologických výpočtech. Například při předpovědi mlhy nebo přízemních mrazů.

Výpočet rosného bodu provádíme podle Clausius-Clapeyronovy rovnice

$$E = E_0 \exp\left[-\frac{L}{R}\left(\frac{1}{T} - \frac{1}{273,16}\right)\right], \quad (1.1)$$

kde E je parciální tlak nasycené vodní páry při teplotě T . Vlastní teplotu rosného bodu z této rovnice získáme tak, že vypočtenou hodnotu E vynásobíme poměrnou vlhkostí. Tuto hodnotu dosadíme zpětně do předchozí rovnice a vypočteme novou hodnotu T . To bude výsledná teplota rosného bodu.

1.5 Měření výšky

Nadmořskou výšku lze měřit více způsoby. Nejprůchoďejší je geodetické měření, kdy vyjdeme od hladiny moře a postupně měříme vzálenosti jednotlivých stanovišť a vertikální úhly, pod kterými je z ostatních stanovišť vidíme. Jednodušší a méně přesné je pak barometrické měření využívající rozdílů atmosférických tlaků. Zde však měření značně komplikuje počasí, díky němuž není ani na jednom a totéž místě tlak stále stejný. Další a asi nejpohodlnější možností je pak použití systému GPS (případně jiné družicové navigace), kdy měřicí přístroj přijme signál z několika družic na obloze a z těchto signálů a z informací dodaných družicemi o jejich aktuální poloze pak vypočte svoji polohu a nadmořskou výšku.

Při měření nadmořské výšky barometrickou metodou se vychází z již zmiňovaného faktu, že s rostoucí výškou klesá atmosférický tlak [10].

Skutečná teplota T se nahrazuje průměrnou teplotou T_s celé atmosféry mezi dvěma hladinami, z nichž v dolní je tlak p_0 a v horní p_z , t_s je průměrná teplota vrstvy v $^{\circ}\text{C}$.

$$T_s = T_n(1 + \alpha t_s) \quad (1.2)$$

$$\alpha = \frac{1}{273,15} \quad (1.3)$$

Změna tíhového zrychlení v závislosti na nadmořské výšce a zeměpisné šířce se stanoví podle rovnice (1.4), kde φ je zeměpisná šířka a h nadmořská výška.

$$g = g_n(1 - 2,644 \cdot 10^{-3} \cos 2\varphi)(1 + 3,14 \cdot 10^{-3} h) \quad (1.4)$$

Laplaceův barometrický vzorec (1.5) slouží k výpočtu výšky nad výchozí hladinou, p_0 je tlak vzduchu ve výchozí hladině, p je tlak ve výšce z , α je konstanta (viz 1.3) a t je teplota vzduchu ve $^{\circ}\text{C}$. Laplaceův vzorec se velice často používá ve zkrácené formě bez členů udávajících změnu tíhového zrychlení.

$$z = 18411(1 + \alpha t) \log \frac{p_0}{p} (1 + 2,644 \cdot 10^{-3} \cos 2\varphi)(1 + 3,14 \cdot 10^{-3} h) \quad (1.5)$$

Za předpokladu, že rozdíl uvažovaných výškových hladin není příliš velký, lze použít Babinetův vzorec (1.6), v kterém t_m je průměrná teplota mezi hladinami ve $^{\circ}C$. Přesnost vzorce je nepřímo úměrná vzdálenosti uvažovaných hladin. Z tohoto důvodu lze tento vzorec uspokojivě použít pouze v případech, kdy vzdálenost obou hladin nepřekračuje 1000 m.

$$\Delta z = 16000(1 + 0,004t_m)\frac{p_0 - p_1}{p_0 + p_1} \quad (1.6)$$

Další možná metoda výpočtu nadmořské výšky je popsána rovnicí 1.7. Tato rovnice je vhodná pro výpočet nadmořské výšky pokud známe výšku výchozího bodu. D je výsledná vypočtená výška, p - tlak na nulové výšce, p_1 - tlak v aktuální výšce, T - průměrná teplota v kelvinech, R - univerzální plynová konstanta 286 a G je tíhové zrychlení 9,81. Průměrná teplota T je považována za teplotu hladiny, v které výšku měříme.

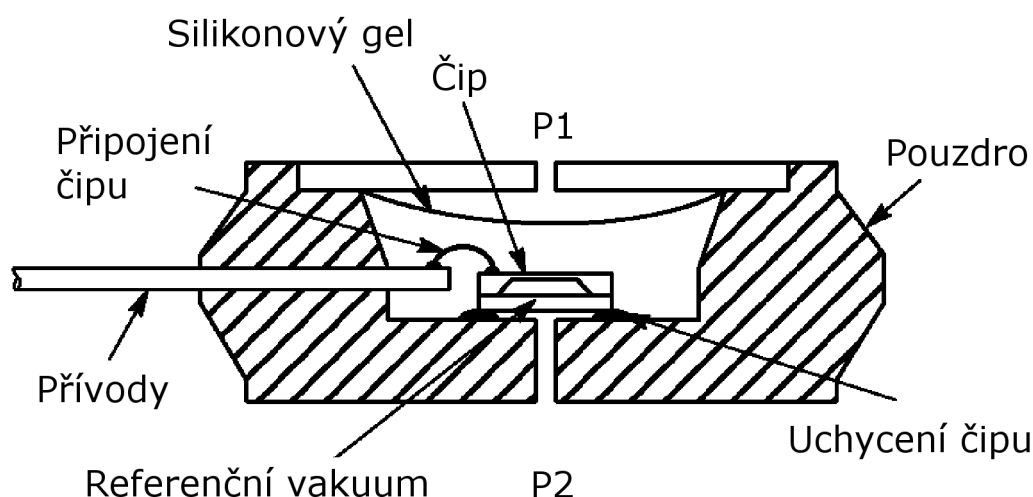
$$D = -\ln \frac{p}{p_1} R \frac{T}{g}, \quad (1.7)$$

Nevýhodou barometrické metody je závislost atmosferického tlaku na počasí. Výškoměr je proto nutné před použitím nastavit podle aktuálních podmínek. To je zvláště důležité v letectví, kde na správném nastavení výškoměru závisí bezpečnost letadla. V turistice se prakticky vždy předem nastavuje taková hodnota, aby výškoměr ukazoval aktuální nadmořskou výšku (známou např. z mapy).

2 SENZORY

2.1 Senzor absolutního tlaku Motorola MPX4115

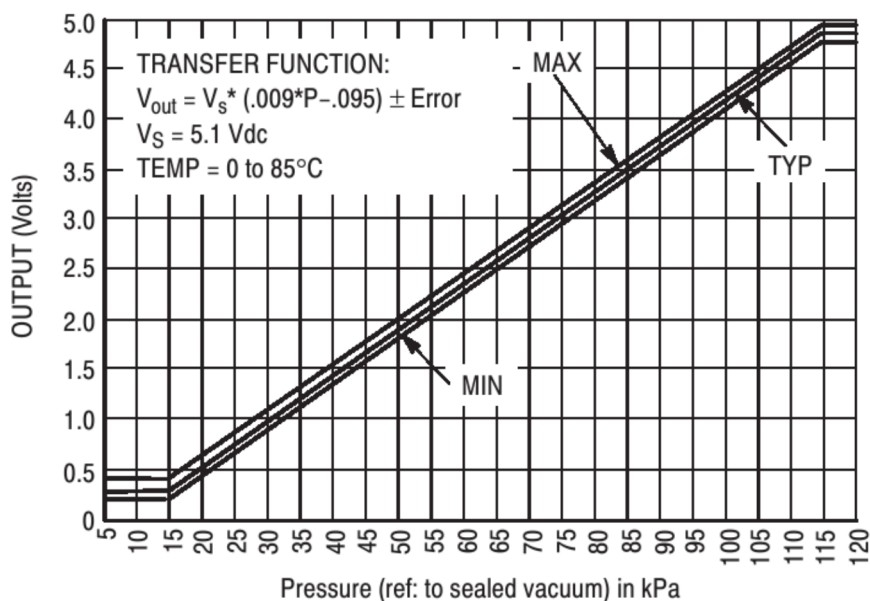
Jedná se o integrovaný snímač tlaku s výstupním zesilovačem a teplotní kompenzací [4]. Senzor měří přímo externí tlak působícího okolního média (plynu) vztažený k nulové hodnotě tlaku vakua ve vnitřní komoře senzoru (viz obrázek 2.1). Tomu odpovídá v klidovém stavu prohnutí stěny čipu rovnající se přibližně tlaku 1 atmosféra (1 ATM) = 14.5 psi. Senzor pracuje na piezorezistivním principu. Čidlo tvoří odporová síť vyrobená na polovodičovém čipu, která se rozvažuje podle mechanického prohnutí čipu.



Obrázek 2.1: Obrázek snímače MPX4115 [4]

Graf na obrázku (2.1) zobrazuje závislost výstupního signálu na tlaku. Typické minimum a maximum křivky je zobrazeno pro teploty v rozsahu 0°C až 85°C. Výstupní napětí je popsáno rovnicí (2.1), kde V_s je napájecí napětí a P je tlak v kPa.

$$V_{out} = V_s(0.009 * P - 0.095) \quad (2.1)$$



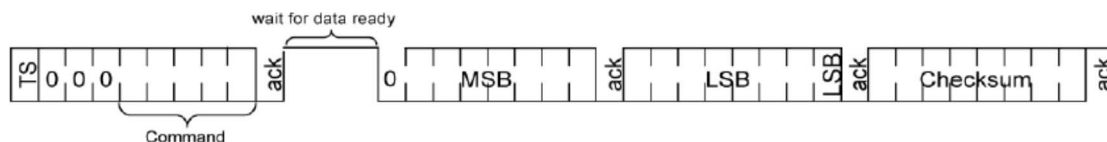
Obrázek 2.2: Závislost výstupního napětí snímače MPX4115 na absolutním tlaku [4]

Tabulka 2.1: Základní parametry snímače MPX4115

Rozsah měření tlaku:	15kPa až 115kPa (2.2 - 16.7 psi)
Teplotní kompenzace:	pro teploty -40 °C až 125 °C
Výstupní napěťový rozsah:	0.2 - 4.8V (offset 0.204)
Maximální chyba:	1.5 % v rozsahu 0 °C až 85 °
Odezva:	1 ms
Napájecí napětí:	+5V
Pracovní teplota:	-40 °C až 125 °C

2.2 Snímač vlhkosti Sensirion SHT11

Jedná se snímač v integrovaném provedení, který má na jednom čipu umístěn jak vlastní senzor, tak i digitální obvody zpracovávající data ze snímačů. V čipu je kromě snímače vlhkosti obsažen také snímač teploty. Digitální část obsahuje kalibrační část, analogově digitální převodník a obvod pro dvoudrátový interface. Sensory jsou kalibrovány v precizní vlhkostní komoře a kalibrační koeficienty jsou uloženy v paměti snímače. Tyto koeficienty jsou využity během měření pro přepočítání údajů ze senzoru. Díky tomu není potřeba snímač jakkoliv kalibrovat. Mezi další výhody snímače patří nízká spotřeba, jednoduché zapojení, dlouhodobá stabilita a malé rozměry. Komunikace je prováděna po dvou vodičové sběrnici a je podobná rozhraní I2C.

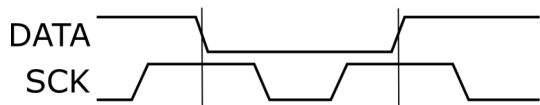


Obrázek 2.3: Ukázka měřící sekvence SHT11

2.2.1 Přenos dat

Pro zahájení přenosu musí být vyslána sekvence "Transmission start" (viz obrázek 2.4). Po startovací podmínce následují vlastní příkazy.

Měřící sekvence začíná buďto příkazem "00000101" pro snímání vlhkosti, nebo "00000011" pro snímání teploty. Kontroler poté čeká než proběhne samotné měření, to trvá 11, 55 nebo 210 ms, podle toho jestli je použito osmi, dvanácti nebo čtrnáctibitové měření. Po skončení měření je signál DATA přepnut na úroveň logické nuly a obvody se přepnou do "idle" módu. Před přečtením vlastních dat musíme počkat na signál "Data Ready". Poté se přenesou dva bajty naměřených dat a jeden bajt kontrolního součtu (CRC). Procesor musí potvrdit každý přenesený bajt přepnutím signálu DATA do nuly.



Obrázek 2.4: Startovací sekvence "Transmission start"

Tabulka 2.2: Příkazy pro komunikaci se snímačem SHT11

Příkaz	kód
Reservováno	0000x
Měření teploty	00011
Měření vlhkosti	00101
Čtení status registru	00111
Zápis do status registru	00110
Reservováno	0101x-1110x
Soft reset	11110

Význam jednotlivých příkazů je patrný z tabulky 2.2. Snad jen k příkazu soft reset dodám, že resetuje zařízení, nastaví status registr na základní hodnoty a je nutno počkat 11ms před dalším příkazem. Popis jednotlivých funkcí status registru je v tabulce 2.3.

Tabulka 2.3: Status registr obvodu SHT11

Bit	Typ	Popis	Implicitně
7		reservováno	0
6	R	Nízké napětí baterie "0" když Vdd je větší než 2,47V "1" když Vdd je menší než 2,47V	X
5		reservováno	0
4		reservováno	0
3		pro testování, nepoužívat	0
2	R/W	vyhřívání	0
1	R/W	kalibrace před měřením	0
0	R/W	"1" = 8bit měření vlhkosti / 12bit měření teploty "0" = 12bit měření vlhkosti / 14bit měření teploty	0

Pro kompenzaci nelinearity výrobce doporučuje [5] použít vztah (2.2), v kterém SO_{RH} je měřená hodnota a koeficienty určuje tabulka 2.4.

$$RH_{linear} = c_1 + c_2 \cdot SO_{RH} + c_3 \cdot SO_{RH}^2 \quad (2.2)$$

Tabulka 2.4: Koeficienty pro linearizaci čidla SHT11

SO_{RH}	c_1	c_2	c_3
12bit	-4	0,0405	$-2,8 \cdot 10^{-6}$
8bit	-4	0,648	$-7,2 \cdot 10^{-4}$

Tabulka 2.5: Technické parametry snímače Sensirion SHT11 pro měření vlhkosti

Parametr	Min.	Typ.	Max.
Rozlišení [%RH]	0,5	0,03	0,03
Rozlišení [bit]	8	12	12
Nelinearita [%RH]		± 3	
Rozsah [%RH]	0		100
Doba odezvy [s]		4	
Hystereze [%RH]		$\pm 0,1$	

Tabulka 2.6: Technické parametry snímače Sensirion SHT11 pro měření teploty

Parametr	Min.	Typ.	Max.
Rozlišení [°C]	0,04	0,01	0,01
Rozlišení [F]	0,07	0,02	0,02
Rozlišení [bit]	12	14	14
Nelinearita [°C]		±0,1	
Rozsah [°C]	-40		123,8
Doba odezvy [s]	5	30	

3 POUŽITÁ ROZHRAŇÍ

3.1 SPI

SPI (serial periperial interface) je velmi rozšířené jednosměrné, nebo obousměrné komunikační rozhraní pro připojování celé řady obvodů k mikroprocesoru. Na rozdíl od paralelního přenosu dat využívá pouze jeden vodič pro data a druhý pro ověření platnosti těchto dat na výstupu. Data jsou přenášena bit po bitu seriově, vysílací strana nikdy nečeká, zda přijímací strana všechna data přijala. Toto je potřeba řešit použitím vhodného komunikačního protokolu. Komunikační rozhraní SPI se používá pro připojování seriových pamětí (RAM, EEPROM), zobrazovacích modulů, A/D převodníků, krokových motorů, nebo procesorů mezi sebou. U mikrokontrolérů AVR lze pomocí SPI programovat vnitřní flash paměť. To umožňuje programovat mikrokontrolér přímo na desce, není třeba jej vyjmout a vkládat do programátoru.

3.1.1 Zapojení SPI

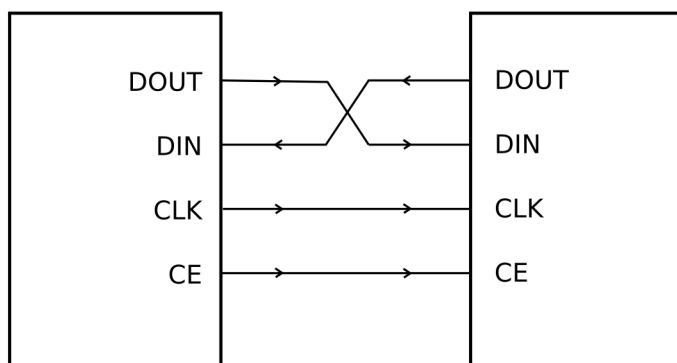
Používá se zapojení dvouvodičové pro jednosměrnou komunikaci, nebo třívodičové pro obousměrnou komunikaci. Tato zapojení bývají vybavena třetím (čtvrtým) vodičem pro výběr periferního obvodu.

Jednotlivé linky jsou:

- MOSI (Master Out Slave In): Výstupní data z master obvodu odeslaná do slave obvodu.
- MISO (Master In Slave Out): Výstupní data z slave obvodu odeslaná do master obvodu.
- SPCK (Serial Clock): Tato řídicí linka je ovládána obvodem master a řídí tok datových bitů. Master může odeslat data s různou přenosovou rychlostí.
- NSS (Slave Select): Řídicí linka vypíná, nebo zapíná slave obvody.

3.1.2 Technické parametry

- Kompatibilita s Embedded 32-bitovými mikrokontroléry
- Komunikace s externími sériovými periferiemi
 - Možnost komunikace s více než 15 zařízeními

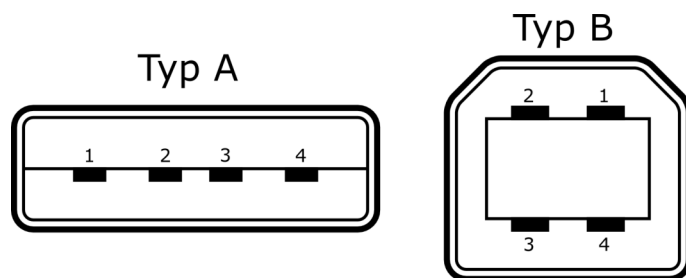


Obrázek 3.1: Schema zapojení SPI pro dva mikroprocesory

- Použití pro seriové paměti (DataFlash, EEPROM)
- Připojení sériových periférií (A/D, D/A převodníky, LCD panely)
- Externí koprocesory
- Komunikace s perifériemi v režimu Master nebo Slave
 - Programovatelná fáze i polarita signálu
 - Programovatelná doba zpoždění mezi hodinovým signálem a daty
 - Programovatelná doba zpoždění mezi následnými přenosy
 - Volba módu detekce chyby

3.2 USB

Rozhraní USB (Universal Serial Bus) [7] je sériová sběrnice nahrazující RS232 (sériový port) a postupně i další rozhraní (LPT, SCSI, GamePort, PS/2). Lze pomocí něj připojovat k počítači většinu dostupných zařízení, od klávesnice po digitální videokamery. Mezi výhody patří možnost připojování a odpojování zařízení za chodu (Plug & Play), vysoká přenosová rychlost (až 480MBit/s), podpora až 127 zařízení a napájení zařízení přímo z USB portu. Na obrázku 3.2 jsou znázorněny dva základní typy konektorů. Typ A se běžně nachází na základní desce počítače, typ B u periferních zařízení. Pro zapojení se používá buď stíněný, nebo nestíněný kabel s kříženými datovými vodiči. Tyto nemusí být krouceny pro přenosové rychlosti do 1,5MBit/s. Napěťové úrovně přibližně odpovídají TTL, pro logickou nulu je to 0,3V, pro logickou jedničku 2,8V.



Obrázek 3.2: USB konektory

Tabulka 3.1: Zapojení USB

Pin	Jméno	Barva	Popis
1	VBus	Červená	+5V DC
2	D-	Bílá	Data -
3	D+	Zelená	Data +
4	GND	Černá	Země

3.2.1 Zapojení USB

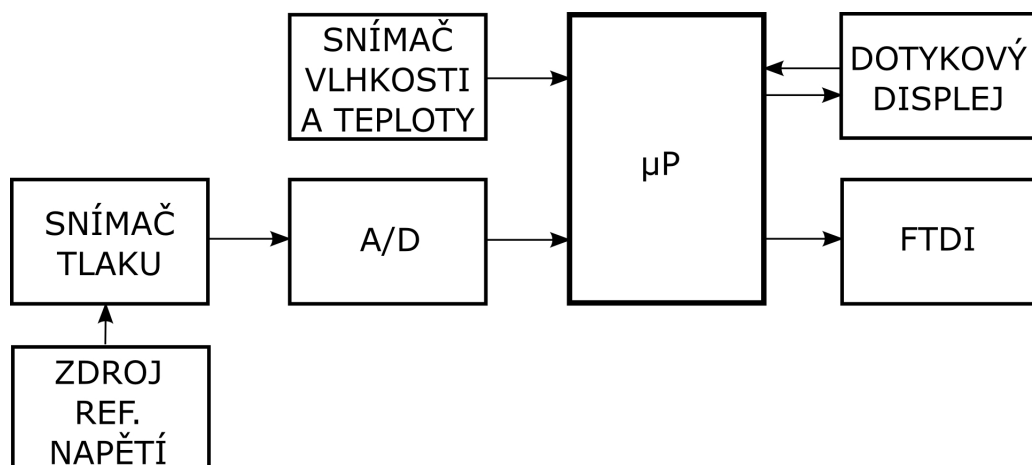
3.2.2 Základní vlastnosti

- Komunikační rychlost od 1,5 Mbit/s do 480Mbit/s
- Rychlosti pro USB v. 2.0:
 - High Speed - 480Mbits/s
 - Full Speed - 12Mbits/s
 - Low Speed - 1.5Mbits/s
- Komunikační vzdálenost do 5m
- Možnost připojení více zařízení
- Rozhraní obsahuje 5V napájení
- Lze připojit až 127 zařízení pomocí jednoho typu konektoru
- USB zajišťuje správné přidělení prostředků (IRQ, DMA, ...)

4 NÁVRH HARDWAROVÉ ČÁSTI

Po snímačích je dalším nejdůležitějším obvodem bezesporu mikrokontrolér, který celé zapojení řídí. Volba vhodného MCU není snadná. Existuje celá řada mikrokontrolérů od 8-mi bitových až po 32-bitové. Liší se jak instrukčními sadami, tak množstvím vstupně-výstupních portů a integrovaných periférií. Moje volba nakonec padla na řadu ATmega od firmy Atmel. Důvodem této volby byl mimo jiné dostupný překladač z jazyka ANSI C zdarma (avr-gcc), existence knihoven avr-libc se základními funkcemi. Další výhodou je možnost programování přímo v obvodu bez nutnosti procesor vyjmout a vložit do programátoru.

Přístroj je tedy řízen osmibitovým mikrokontrolérem ATmega64 z rodiny AVR firmy Atmel[3], na jehož porty jsou připojeny snímače, displej a ftdi čip pro přenos dat do počítače přes rozhraní USB. To je znázorněno na blokovém schématu 4.1. Kromě znázorněných bloků zapojení obsahuje pouze napájecí obvody a pár pasivních součástek.



Obrázek 4.1: Blokové schéma zapojení

4.1 Procesor Atmel ATmega64a

Mikrokontrolér byl vybrán s ohledem na dostatečný počet vstupně-výstupních portů, výkon a dostatek paměti EEPROM pro sběr dat. V neposlední řadě také pro dostupnost kvalitní dokumentace a příznivou cenu. Základní vlastnosti jsou:

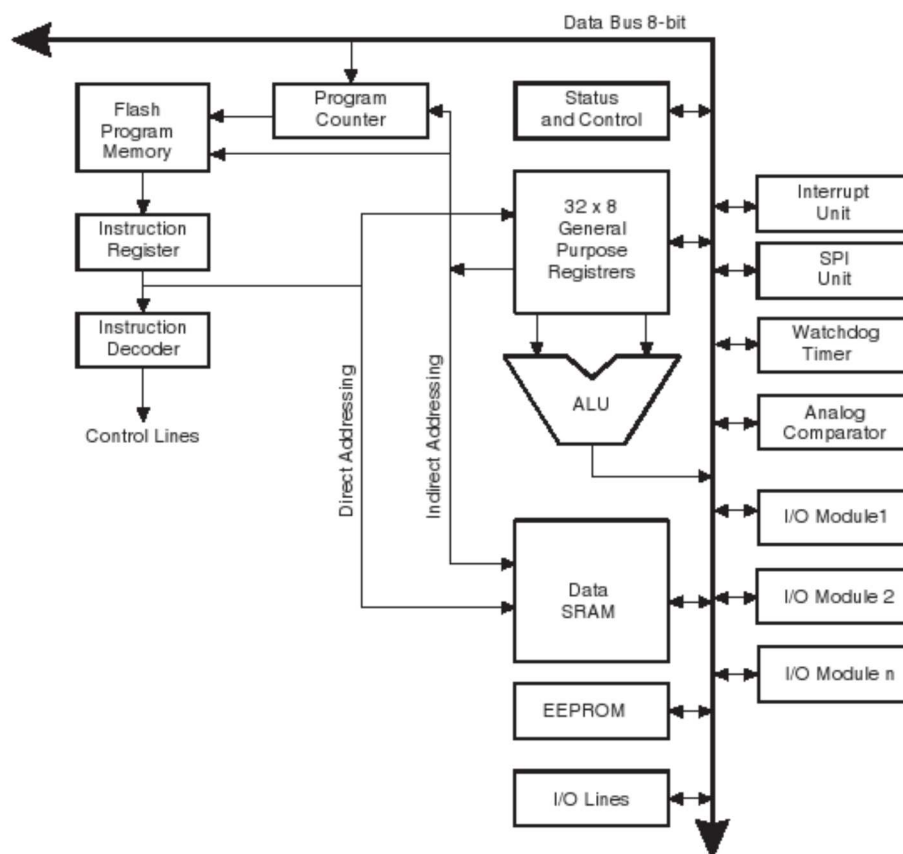
- vysoký výkon - až 16 MIPS při 16MHz
- 130 instrukcí, z nichž se většina provede v jednom hodinovém cyklu
- 64kB flash paměti s výdrží 10000 přepisů

- 2kB EEPROM paměti s výdrží 100000 přepisů
- 4kB interní SRAM paměť
- možnost adresace až 64kB externí paměti
- programovatelný zámek pro zabezpečení softwaru
- SPI rozhraní pro programování přímo v aplikaci
- JTAG rozhraní pro programování a ladění
- dva 8mibitové čítače/časovače
- dva rozšířené 16bitové čítače/časovače
- dva 8mibitové PWM kanály
- 8mikanálový 10b A/D převodník
- bitově orientované dvouvodičové sériové rozhraní
- dvě programovatelná USART rozhraní
- SPI rozhraní
- Watchdog
- 53 programovatelných V/V linek
- napájení 4.5 až 5.5 V
- rychlost 0 - 16 MHz
- vnitřní RC oscilátor

Jádro procesoru

Procesor ATmega64 je založen na RISCovém jádru AVR Harvardské architektury, jehož blokové schéma je na obr. 4.1. To se stará o provádění programu, přístupu k pamětem a periferiím a obsluhu přerušování. Pro zrychlení provádění instrukcí má jednoúrovňovou pipeline, kdy během provádění jedné instrukce se následující instrukce načítá z programové paměti.

ALU (aritmeticko-logická jednotka) má přímý přístup k 32 8-bitovým GPR (registrům pro všeobecné použití) uspořádaných do 8 banků. Během jednoho cyklu ALU načte až 2 operandy z GPR, provede výpočet a výsledek uloží zpět do GPR. 6 8-bitových GPR registrů může být použito jako 3 16-bitové registry (X, Y, Z) pro



Obrázek 4.2: Blokový diagram AVR architektury [3]

nepřímé adresování datové paměti a jeden z nich i pro adresaci programové Flash paměti (pro přístup k různým LUT tabulkám, konstantám, a pod.). ALU podporuje standardní aritmetické, logické a bitové operace. Některé AVR procesory, jako např. ATmega navíc mají i hardwarovou násobičku pro celočíselnou a desetinnou aritmetiku (výpočet trvá 2 cykly). Po provedení operace ALU nastaví příslušné flagy ve stavovém registru.

Běh programu (registr program counter) lze řídit podmíněnými a nepodmíněnými skoky, voláním a návratem z podprogramu a voláním obsluh interních či externích přerušení.

Programová i datová paměť má lineární adresování. Programová paměť Flash se dělí na oblast aplikační a oblast zavaděče (bootcode). Každá oblast má svoji pojistku proti čtení/zápisu. V zaváděcí oblasti může být uložen program zavaděče, který přes nějaké rozhraní přijme aplikační program a pomocí instrukce SPM (nelze ji volat programem z aplikační oblasti) jej zapíše do aplikační paměti a pak mu předá řízení. Tento mechanismus lze využít pro programování i přes jiné rozhraní, než k tomu určené SPI/JTAG.

Zásobník sdílí interní paměť SRAM a roste směrem dolů. Před voláním podprogramů nebo obsluh přerušení je nutné nastavit ukazatel zásobníku (SP) tak, aby byl k dispozici dostatek paměti pro návratové adresy. Každé přerušení lze individuálně nebo globálně povolit/zakázat. Priorita přerušení je daná adresou vektorů přerušení (čím nižší adresa vektoru, tím vyšší priorita).

Programovatelné konfigurační pojistky

Procesor obsahuje kromě paměti flash a EEPROM ještě čtyři osmibitové paměťové buňky kterým se říká pojistkové (fuses). Tyto čtyři registry se programují zvlášť. Konfigurační pojistky se zpravidla programují pouze jednou, protože slouží k základnímu nastavení procesoru. Pokud je pojistka naprogramovaná, má daný bit hodnotu **0**, pokud naprogramovaná není, má hodnotu **1**. Při programování pojistek je potřeba opravdu pečlivě zkontrolovat co nastavujeme, jelikož nevhodná konfigurace může mít za následek to že již s procesorem nebudeme moci nadále pracovat. Dojít k tomu může například pokud nastavíme špatnou hodnotu externího krystalu a nebo ještě hůře pokud zamkneme procesor proti dalšímu přeprogramování.

Nižší byte programovatelných pojistek má následující význam:

- **BODLEVEL** - určuje napěťovou úroveň při které dochází k vyvolání resetu. Pro hodnotu 1 je to 2,7 V a pro 0 je to 4,0 V (od výrobce 1)
- **BODEN** - zapíná detektor podpětí (od výrobce 1)
- **SUT1, SUT0** - nastavuje přídavné časové zpoždění rozběhu procesoru po zapnutí napájecího napětí (od výrobce 1, 0)
- **CKSEL3..CKSEL0** - nastavuje druh a kmitočet oscilátoru. Defaultní nastavení je na interní RC oscilátor s kmitočtem 1MHz. Další nastavení je patrné z tabulky 4.2 (od výrobce 0, 0, 0, 1). Například pro zapnutí interního oscilátoru na kmitočtu 8 MHz je potřeba nastavit CKSEL na 0100.

Tabulka 4.1: Fuse Low

BODLVL	BODEN	SUT1	SUT0	CKSEL3	CKSEL2	CKSEL1	CKSEL0
--------	-------	------	------	--------	--------	--------	--------

Vyšší byte programovatelných pojistek má následující význam:

- **OCDEN** - povoluje funkci On-Chip debug pro ladění přes rozhraní JTAG (od výrobce 1)
- **JTAGEN** - zapne rozhraní JTAG (od výrobce 0)

Tabulka 4.2: Nastavení hodinového signálu

Nastavení časování	CKSEL3..CKSEL0
Externí krystal	1111 - 1101
Externí krystal na nízkém kmitočtu	1001
Externí RC oscilátor	1000 - 0101
Kalibrovaný interní RC oscilátor	0100 - 0001
Externí hodiny	0000

- **SPIEN** - zapne rozhraní SPI (od výrobce 0)
- **CKOPT** - nastavuje režim oscilátoru, pokud je zapnut budič krystalového oscilátoru bude používat plný rozkmit napětí. To je zapotřebí pokud je použit rychlejší krystal než 8 MHz. Dochází však k zvýšení spotřeby. (od výrobce 1)
- **EESLAVE** - pokud je naprogramován, paměť EEPROM je chráněna proti vymazání. To zabrání například vymazání při programování flash paměti. (od výrobce 1)
- **BOOTSZ1, BOOTSZ0** - nastavuje velikost zaváděcí (boot) oblasti v rozmezí 512 až 4096 B (od výrobce 0, 0)
- **BOOTRST** - nastavuje vektor resetu - pokud je bit naprogramovaný, spustí se po resetu zaváděč ze zaváděcí oblasti, jinak se spustí program z aplikační oblasti (od výrobce 1)

Tabulka 4.3: Fuse High

OCDEN	JTAGEN	SPIEN	CKOPT	EESAVE	BSZ1	BSZ0	BRST
-------	--------	-------	-------	--------	------	------	------

Rozšiřující konfigurační pojistky obsahují pouze dva nastavitelné bity a to **M103C** který slouží k zapnutí módu kompatibility s mikrokontrolérem ATmega103. V tomto módu jsou některé periferie ATmega64 vypnuty, aby byl zpětně kompatibilní s ATmega103. Defaultně je tento pin naprogramován, pro naše použití je třeba jej vypnout. Další bit (**WDTON**) znemožňuje vypnutí obvodu wtchdog. Od výrobce není programován.

Tabulka 4.4: Extended Fuse

-	-	-	-	-	-	M103C	WDTON
---	---	---	---	---	---	-------	-------

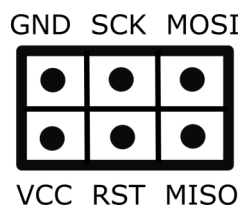
Bity **LB1** a **LB2** v registru Fuse Bits slouží k zakázání zápisu do flash a EEPROM paměti přes paralelní, nebo SPI/JTAG rozhraní. Bity **BLB12**, **BLB11**, **BLB02** a **BLB01** slouží k nastavování zaváděcí oblasti a jako zámky uživatelského programu. Všechny fuse bity jsou implicitně nastaveny do jedničky. Pro moje zapojení je třeba JTAGEN vypnout, aby bylo možné použít piny PF4 a PF5 jako výstupy.

Tabulka 4.5: Lock Bits

-	-	BLB12	BLB11	BLB02	BLB01	LB2	LB1
---	---	-------	-------	-------	-------	-----	-----

Programování přes ISP

ISP (In System Programming) [3] umožňuje programování mikrokontroléru přímo ve výsledné aplikaci bez nutnosti vyjmát vlastní součástku a vkládat do programátoru. Stačí poze na desku plošného spoje umístit konektor s vyvedenými signály MISO, MOSI, SCK a RST popřípadě ještě napájení a GND. **Pozor!** u mikrokontroléru ATmega64 **NEJSOU** pro ISP použity signály označené MISO a MOSI, jak je tomu u většiny obvodů rodiny AVR ale piny PDO a PDI!

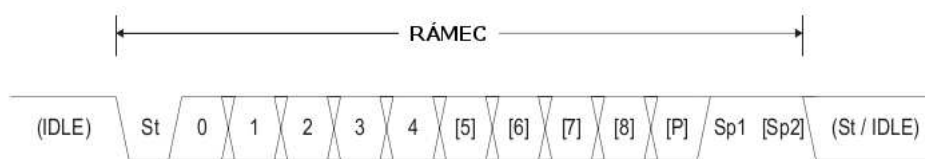


Obrázek 4.3: Signály na ISP konektoru výškoměru

USART

Procesor ATmega64 obsahuje dva obvody USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter). Jedná se o obvody umožňující plně duplexní synchronní, nebo asynchronní sériový přenos. Umožňuje přenos rámců s 5 až 9 datovými bity a jedním nebo dvěma stop bity. Může generovat sudou nebo lichou paritu s hardwarovou kontrolou. Obvod USART se skládá z generátoru hodinového signálu obstarávajícího synchronizaci vysílání i přijímání v synchronním režimu, vysílacích a přijímacích obvodů.

Rízení sériového kanálu slouží tyto registry:



Obrázek 4.4: USART - Formát rámce

- **UDR** - datový registr obsahující odeslanou nebo přijatou hodnotu, fyzicky je představován dvojicí registrů. Při zápisu do UDR se naplní vysílací registr, při čtení s UDR se čte z přijímacího datového registru.
- **USR** - registr obsahující stavové příznaky přenosu:
 - **RXC** - příjem kompletní
 - **TXC** - vysílání kompletní
 - **UDRE** - datový registr je prázdný
 - **FE** - chyba rámce
 - **OR** - ztráta znaku
- **UCR** - řídicí registr UART
- **UBRR** - registr přenosové rychlosti

$$PR = \frac{f_O}{16(UBRR + 1)}, \quad (4.1)$$

kde PR je přenosová rychlost v Bd, f_O kmitočet krystalu a $UBRR$ obsah registru $UBRR$.

4.2 Snímací obvody

4.2.1 Měření tlaku

Pro měření tlaku byl zvolen snímač absolutního tlaku MPX4115 [4]. Jako zdroj referenčního napětí byl zvolen obvod ANALOG DEVICES REF02. Výstupní napětí snímače nabývá hodnot od 4,725 až 4,863V s citlivostí 45.9mV/kPa. Pokles tlaku o 1hPa odpovídá změně výšky o 8m. Podobně změna výšky o 1m vyvolá změnu výstupního napětí čidla o 0,6mV.

4.2.2 Měření vlhkosti

Zapojení snímače vlhkosti [5] je velmi jednoduché, stačí přivést napájecí napětí přes blokovací kondenzátor, zem a připojit dvěma vodiči k mikrokontroléru. Komunikační protokol je popsán výše (viz kap. 2.2).

4.3 A/D převodník MCP3202

Abychom mohli připojit snímač tlaku na mikrokontrolér, je použit A/D převodník Microchip MCP3202. Jedná se o 12ti bitový dvoukanálový A/D převodník s SPI (viz kap. 3) rozhraním a následujícími základními parametry:

- 12bitové rozlišení
- 2 kanály
- diferenciální a integrální nelinearita ± 1 LSB
- analogové vstupy použitelné jako SE (měření proti zemi) nebo jako pseudodiferenční pár
- na čipu je zabudovaný obvod S/H (Sample and Hold, vzorkovač)
- napájení 2,7 až 5,5 V
- maximální vzorkovací rychlost je 200 000 vzorků za sekundu pro $U_{CC} = 5$ V a 50 000 vzorků za sekundu pro $U_{CC} = 2,7$ V
- použitá nízkopříkonová CMOS technologie (klidový proud až 5 nA, v aktivním režimu 0,55mA při $U_{CC} = 5$ V)
- průmyslový rozsah pracovních teplot (-40°C $+85^{\circ}\text{C}$)

Vstupy jednotlivých kanálů CH0 a CH1 mohou být konfigurovány buď jako nezávislé kanály v režimu SE, kdy se napětí měří proti zemi, nebo do pseudo-diferenčního režimu s možností volby orientace (CH0 - CH1, nebo CH1 - CH0). $\overline{CS}/SHDN$ vybírá režim obvodu. Je-li v log. 1 je obvod vypnut a má sníženou spotřebu. Je-li v log. 0 je obvod připraven na vyslání převedených dat. CLK je používán pro začátek převodu a synchronizaci převáděných dat.

Výstupní kód obvodu na výstupu D_{OUT} je závislý na vstupním napětí a velikosti referenčního napětí. Je-li vstupní napětí menší než 0 V, je vrácena hodnota 0x000. Je-li větší než $U_{REF} - 1\text{LSB}$, je vrácen kód 0x3FF. Výstupní digitální kód je dán vzorcem:

$$vystupnikod = \frac{4096}{U_{REF}} U_{VST} \quad (4.2)$$

Tabulka 4.6: Význam konfiguračních bitů MCP3202

č	Název	Význam
1	start-bit	začátek komunikace
2	SGL/\overline{DIFF}	volba režimu vstupů: 1-SE, 0-pseudodiferenční
3	ODD/\overline{SIGN}	volí kanál nebo polaritu: $SGL/\overline{DIFF} = 1$, volí kanál: 0 - CH0, 1 - CH1 $SGL/\overline{DIFF} = 0$, volí kanál: 0 - CH0 - CH1, 1 - CH1 - CH0
4	$MSBF$	volí způsob vysílání: MSBF=1 vysílá v pořadí MSB až LSB a doplní nulami MSBF=0 vysílá v pořadí LSB až MSB a potom LSB až MSB

4.4 Napájení

Zapojení je napájeno třemi AAA bateriemi. Protože by napětí baterií (3,6-4,5V) nestačilo pro některé obvody, jsou použity dva DC/DC měniče MAX8570.

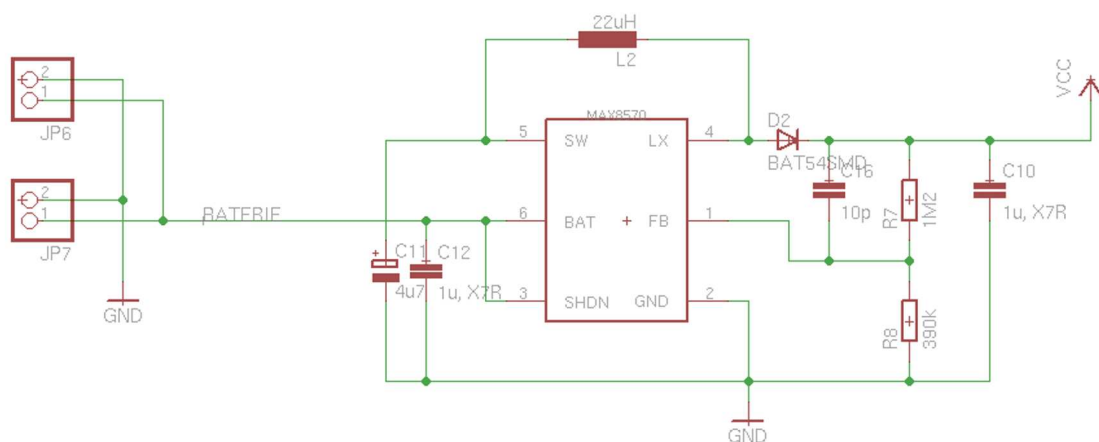
Obvod MAX8570 může mít na výstupu nastaven rozsah napětí od velikosti napájecího napětí až po 28V. Hodnota výstupního napětí se nastavuje odporovým děličem mezi piny 4 a 6. Velikost odporů lze vypočítat podle rovnice:

$$R_1 = R_2 \left(\frac{V_{OUT}}{V_{FB}} - 1 \right), \quad (4.3)$$

kde V_{OUT} je velikost výstupního napětí, $V_{FB} = 1,226V$, R_1 a R_2 je výsledný odporový dělič. Děličem $R_1 = 2M2$, $R_2 = 390k$ je výstupní napětí prvního měniče nastaveno na hodnotu $V_{OUT} = 8,14V$, což je napětí potřebné pro napájení obvodu reference pro snímač tlaku. Děličem $R_1 = 1M2$, $R_2 = 390k$ je výstupní napětí druhého měniče nastaveno na hodnotu $V_{OUT} = 5V$. První měnič je zapnut stále a napájí mikrokontrolér a displej. Druhý měnič napájí snímač tlaku a je spínán pouze v okamžiku snímání hodnoty, kvůli úspoře energie.

Přestože byl napájecí obvod navrhován pro dvě tužkové baterie, není to nejvhodnější řešení. Při zapnutí přístroje měnič začne odebírat proud z baterií. Ukázalo se, že ne pokaždé je tento proud stejný. Stává se že občas měnič baterie zatíží tak, že jejich napětí poklesne pod úroveň 2.7 V což je minimální napájecí napětí a měnič se nezapne správně. Řešením je použití li-ion akumulátoru, který má vyšší napájecí napětí (3.6V) a větší proudovou zatížitelnost.

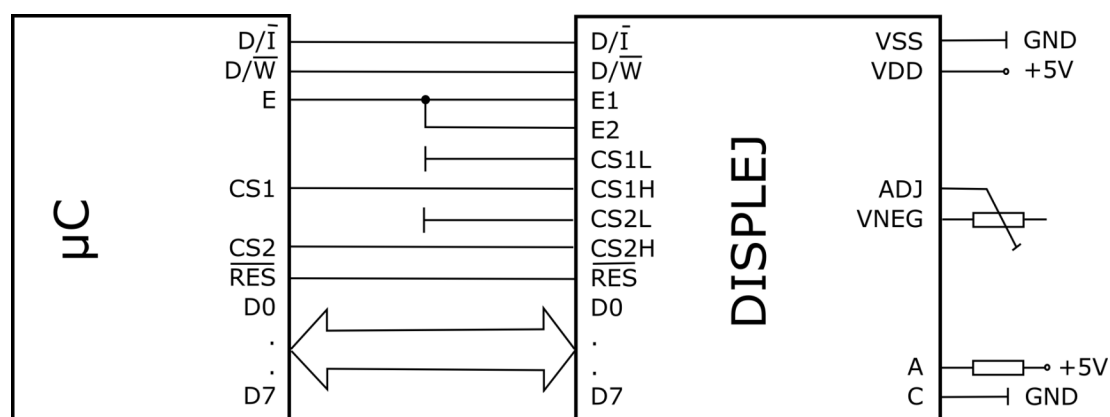
Odběr zařízení je poměrně nízký. Spotřeba se pohybuje mezi 40 mA při normálním provozu až 70 mA při zapnutém podsvícení displeje.



Obrázek 4.5: Zapojení napájecího obvodu

4.5 Displej

Komunikace s uživatelem probíhá pomocí grafického dotykového displeje Electronics Assembly DIP128-6 [6]. Rozlišení displeje je 128 x 64 bodů. Zobrazované body mají černou barvu, podsvícení je žluto-oranžové. Displej je vybaven řadičem KS0108. Data mezi procesorem a řadičem jsou přenášena po osmibitové paralelní sběrnici, pro ovládání displeje jsou dále zapojeny vývody D/I, RW, E, CS1L, CS2L, jejich význam je popsán v tabulce 4.7. Výsledné zapojení je patrné z obrázku 4.6. Pro úsporu energie jsem zapojil podsvícení přes tranzistor, kterým bude spínáno pouze v okamžiku, kdy bude potřeba přístroj ovládat.



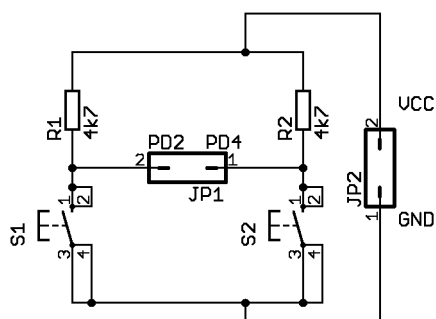
Obrázek 4.6: Schéma zapojení displeje

Displej je vybaven dotykovým panelem. Ten je maticový s třemi řádky a pěti sloupci. Osmi vodiči je připojen přímo na port D mikroprocesoru. Pozice stisklého pole lze vyhodnocována skenováním řádků a sloupců matice. Bohužel během ožívování

Tabulka 4.7: Zapojení displeje DIP128-6

Vývod	Název	Funkce
1	VSS	napájení GND
2	VDD	napájení +5V
3	ADJ	nastavení kontrastu
4	RES	reset ("0" = reset)
5	D/I	volba mezi daty a instrukcemi ("0" = instrukce)
6	R/W	volba mezi zápisem a čtením ("0" = čtení)
7	E1	zapnutí levé části displeje
8	E2	zapnutí pravé části displeje
9	CS1L	výběr levé části ("0")
10	CS1H	výběr levé části ("1")
11	CS2L	výběr pravé části ("0")
12	CS2H	výběr pravé části ("1")
13	VNEG	nastavení kontrastu
14	NC	nepřipojeno
15 - 22	D0 - D7	data
23	LED+	podsvícení displeje +
24	LED-	podsvícení displeje -

přípravku se ukázalo že materiál kontaktů není dostatečně kvalitní na to, aby vydržel několik připojení a odpojení. Kontakty se opotřebily (asi jen po dvou odpojeních) natolik že se používání dotykového panelu stalo velmi nestabilní. Proto jsem nakonec tento panel nepužil a nahradil jej dvěma tlačítky, které budou pro ovládání přístroje plně postačující.



Obrázek 4.7: Schéma zapojení desky s tlačítky

4.6 Komunikace po USB

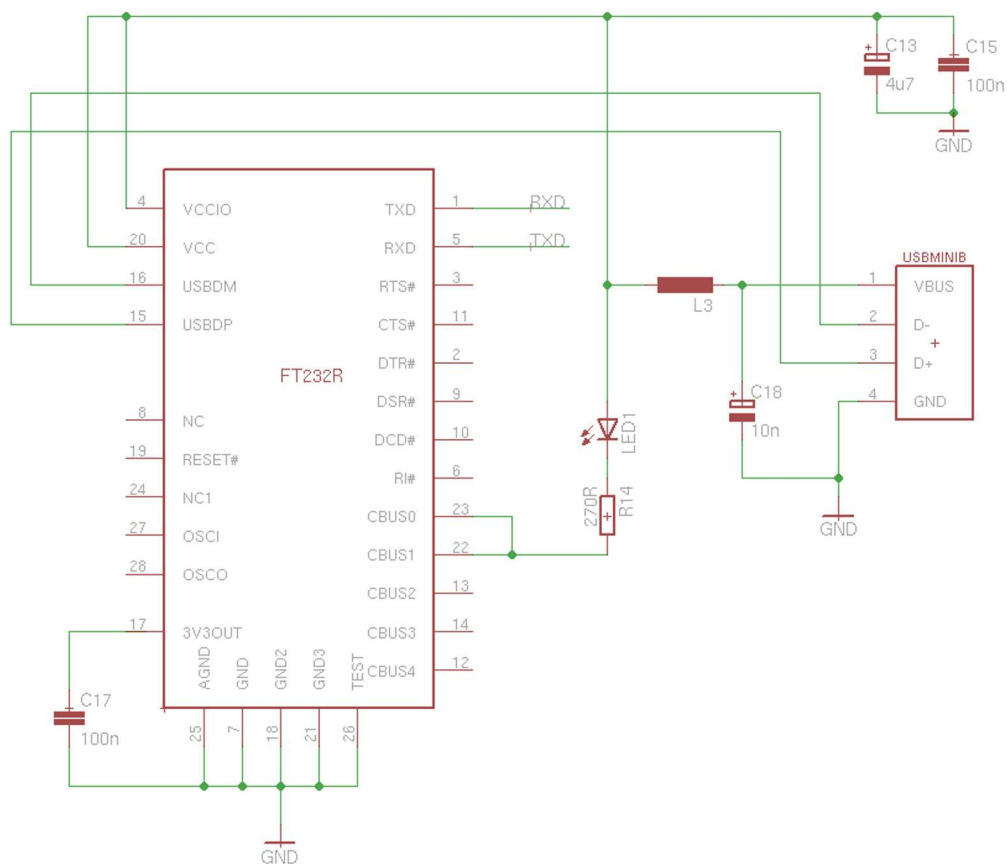
Pro přenos naměřených dat do počítače jsem zvolil USB rozhraní (viz kap. 3.2). To je implementováno obvodem FTDI FT232R [12]. Jedná se o převodník USB na UART s volitelným generováním hodinového signálu a s hardwarovým bezpečnostním klíčem. Obvod přímo integruje EEPROM paměť, hodinové obvody a USB residtory přímo na čipu. Vnitřně generovaný hodinový signál (6MHz, 12MHz, 24MHz, a 48MHz) je vyveden z čipu a může být použit například pro mikrokontrolér nebo jiné externí obvody. Unikátní identifikační číslo je zapsáno do obvodu při výrobě a je čitelné přes USB. Toto číslo je využíváno pro hardwarové zabezpečení a může být použito pro zabezpečení uživatelské aplikace proti zkopírování. Výhodou obvodu je potřeba minima externích součástek. Obvod má implementován USB protokol přímo na čipu. Základní parametry jsou:

- UART rozhraní podporuje 7 nebo 8 datových bitů, 1 nebo 2 stop bity, sudou, lichou, popřípadě žádnou paritu
- plně ovládané hardwarové nebo softwarové (X-On/X-Off) řízení toku
- rychlost přenosu od 300 baud až do 3 Magabaud
- buffer pro odesílaná (256 B) a přijímaná (128 B) data
- volně dostupné VCP a D2XX ovladače
- integrovaný zdroj hodinového signálu

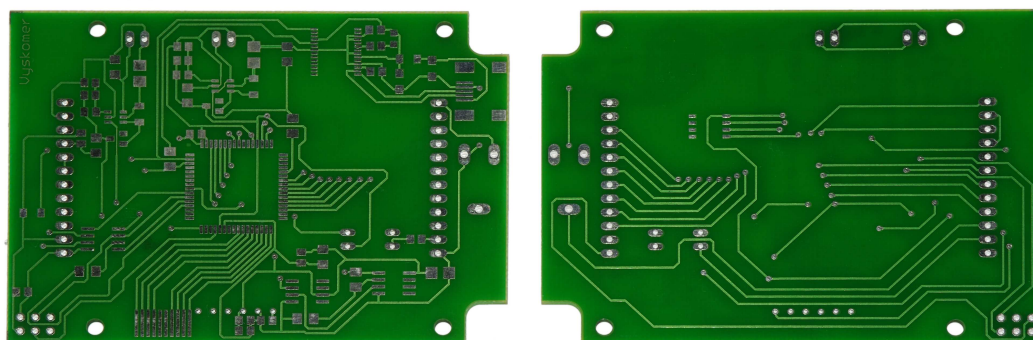
Obvod je sériovou linkou (Rx/Tx) připojen k mikrokontroléru. Z důvodu snadnějšího ladění jsou vývody CBUS1, CBUS2 a připojeny na LED diodu, pro indikaci přenosu dat. Na čipu je obsažen interní oscilátor na frekvenci 12MHz. Proto zapojení nevyžaduje externí krystal. Napájení obvodu je realizováno přímo z USB, čímž šetříme baterie.

4.7 Mechanické provedení

Tištěný spoj je navržen jako oboustranný, prokovený s rozměry 62x85mm. Veškeré součástky jsou osazeny na vrchní straně desky. Na spodní straně desky je umístěn pouze snímač tlaku. Deska je spolu s displejem umístěna v plastovém pouzdru s prostorem pro tři AAA baterie. Na pravé straně krabičky je vyveden USB mini konektor typu B a hlavní vypínač přístroje. Na bočnici pod displejem jsou umístěna dvě tlačítka sloužící k ovládání a kalibraci.



Obrázek 4.8: Zapojení obvodu FT232R



Obrázek 4.9: Neosazená deska tištěného spoje

5 NÁVRH SOFTWARE ČÁSTI

V této kapitole zmíním koncepci řídicího programu a podrobněji rozeberu důležité funkce použité v programu. Celý kód programu je napsán v jazyce ANSI C. Pro překlad jsem použil avr-gcc, což je překladač jazyka C pro AVR mikrokontroléry. Program se skládá z hlavního souboru (vyskomer.c) obsahujícího vlastní kód programu, ze souborů s funkcemi pro displej (ks0108.c) a snímač vlhkosti (sensirionprotocol.c) a několika hlavičkových souborů (font12x16.h, font6x8.h, ks0108.h a sensirionprotocol.h).

Z vývojového diagramu (Příloha. C) je patrná základní funkce programu. Po zapnutí přístroje na uživatele čeká několik dotazů. Nejprve je to možnost přenést data uložená z předchozích měření do počítače. Další dialog slouží k nastavení kalibračních hodnot které si program uloží pro další použití. Pokud nebude nastavena kalibrační hodnota, bude výpočet vycházet z normálního tlaku. Posledním otázkou před započítáním měření je volba zda budou měřená data ukládána do paměti či nikoliv. Po projití této nastavovací procedury následuje začátek vlastního měření. První věcí, která se provede je zapnutí snímačů, následuje odečtení napětí ze snímače tlaku a teploty. V dalším kroku se opět vypnou snímací obvody a přejde se k vlastnímu výpočtu. Provede se výpočet teploty a tlaku. Z těchto hodnot se dále dopočítává nadmořská výška. Pokud byla při spuštění provedena kalibrace, bude se při výpočtu brát v potaz. Jakmile dojde k dopočítání všech potřebných hodnot, vypíše se na displej údaje o aktuální teplotě, tlaku a nadmořské výšce, popřípadě informace o zbývajícím paměti. Pokud bylo zapnuto ukládání, zapíše se údaj o aktuální nadmořské výšce do paměti. V dalším kroku se opět program vrací k zapnutí snímačů a následuje další cyklus měření. Pokud se provádí záznam do paměti, program hlídá zda nebyla paměť již zaplněna.

5.1 Konstanty a makra

Na začátku zdrojového kódu je několik užitečných maker a definice konstant. Jelikož přístup k jednotlivým vstupně-výstupním pinům je v ANSI C celkem nepřehledný a složitý na zápis, nadefinoval jsem si čtyři makra pro nastavení, vynulování, invertování a přečtení bitu.

```
#define SETB(data,bitnum) (data|=(1<<bitnum)) // nastav bit
#define CLRB(data,bitnum) (data&=~(1<<bitnum)) // vynuluj bit
#define FLIPB(data,bitnum) (data^=(1<<bitnum)) // invertuj bit
#define GETB(data,bitnum) ((data>>bitnum)&1) // precti bit
```

Následují další makra pro jednodušší práci s tlačítky připojeným na pin 4 a 7 portu D.

```
#define TL1 (PIND << (1<<PIND4))  
#define TL2 (PIND << (1<<PIND7))
```

5.2 Komunikace s A/D převodníkem

Vlastní čtení dat z A/D převodníku zprostředkovávají dvě funkce a to funkce `SPI_MasterInit()`, která nastaví piny MOSI a SCK jako výstupní, ostatní jako vstupní. Dále povolí SPI u procesoru v módu master a nastaví frekvenci hodinového signálu SPI na šestnáctinu frekvence procesoru.

```
void SPI_MasterInit(void)  
{  
    // Nastavi MOSI a SCK jako vystup  
    DDRB = (1<<PB2)|(1<<PB1) | (1<<PB0);  
    // Povolí SPI, Master, nastavi hodiny na fck/16  
    SPCR = (1<<SPE)|(1<<MSTR)|(1<<SPR0);  
}
```

Další funkcí pro SPI komunikaci je `SPI_MasterTransmit()` která obstarává odeslání osmi bitů do registru v A/D převodníku a hned poté přijímá 8 bitů z převodníku.

```
char SPI_MasterTransmit(char cData)  
{  
    // Zacatek prenosu  
    SPDR = cData;  
    // Pocka na konec prenosu  
    while(!(SPSR & (1<<SPIF)));  
    return SPDR;  
}
```

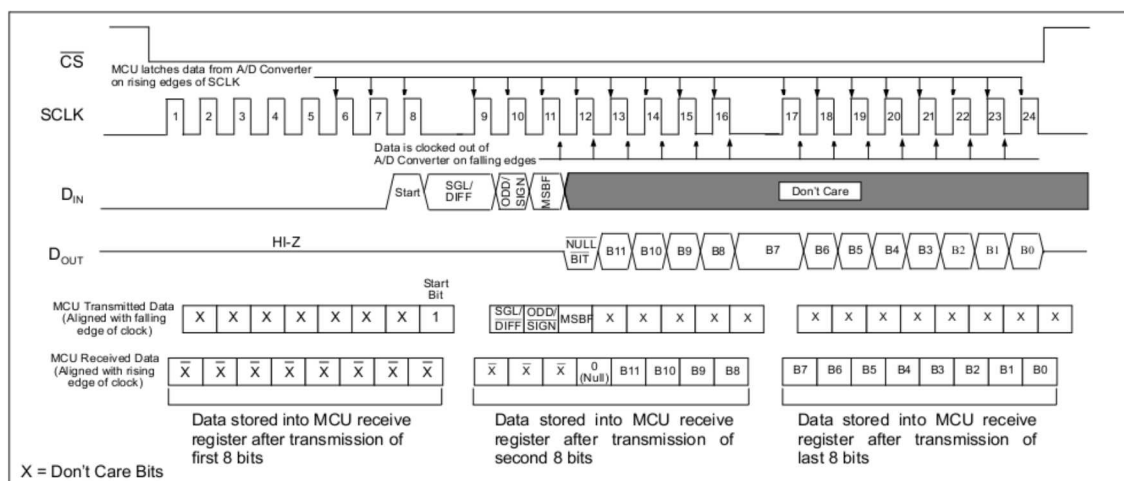
Průběh vlastní komunikace je patrný z obrázku 5.1. Před začátkem komunikace je zapotřebí nastavit signál CS na úroveň log. 0. Následuje odeslání osmi bitů obsahujících start-bit (0x01). Následující přijímaný bajt nás nezajímá. Na začátku další osmibitové sekvence (0xE0) se odešlou tři řídicí bity kterými se volí režim vstupu, nastavuje kanál a formát dat (viz. 4.6). Po odeslání druhého bajtu se hned přijímá bajt obsahující první čtyři bity měřené hodnoty. Další odesílaný bajt může obsahovat libovolná data, po jeho odeslání se přijímá zbylých 8 bitů měřené hodnoty. Na závěr komunikace opět se signál CS přepne zpět do log. 1.

```

SPI_MasterInit();
AD=0;
PORTB&=~(1<<PB0);
SPI_MasterTransmit(0x01);
*((((unsigned char*)&AD)+1)=(0x0F & SPI_MasterTransmit(0xE0)));
*((unsigned char*)&AD)= (SPI_MasterTransmit(0x00));
PORTB|=(1<<PB0);

```

Pro zvýšení přesnosti měření každé měření hodnoty vkládám do cyklu, odečítám hodnotu 1024krát a vytvářím průměr. Tímto jsou eliminovány přechodové jevy vznikající odečtem hodnot ležících na hraně dvou hladin, které je schopný A/D převodník rozpoznat.



Obrázek 5.1: SPI komunikace po osmibitových segmentech [11]

5.3 Výpočet hodnot tlaku, teploty a nadmořské výšky

Ve chvíli kdy mám uloženou průměrnou hodnotu z 1024 odečtených hodnot, následuje vlastní výpočet atmosferického tlaku. Hodnotu z převodníku je třeba vynásobit napájecím napětím a vydělit počtem vzorků ($2^{12} = 4096$). Tím dostaneme napětí na výstupu snímače tlaku. To dosadíme do rovnice 2.1 udávané výrobcem snímače tlaku. Oba zmiňované úkony provádím v jednom kroku.

```

tlak=((((float)AD*5)/4096)/5+0.095)/0.009)*10;

```


Pro odečtení teploty a vlhkosti ze snímače používám upravené funkce doporučené výrobcem čidla. Tyto funkce se nacházejí v souboru (`sensirion_protocol.c`). Nejprve zavolám funkci `s_measure`, jako parametry ji předávám adresu proměnné, do které se má uložit hodnota změřené teploty (`teplota_raw`) a jako druhý parametr číslo "0", což znamená požadavek na změření teploty. Návrátovou hodnotou je výsledek výpočtu kontrolního součtu. Pokud se výsledek rovná nule, data byla bezchybně přenesena. Stejnou funkcí mohu odečíst aktuální vlhkost, pouze ji jako parametry předám adresu proměnné `vlhkost_raw` a jako druhý parametr dám číslo "1", což je požadavek na měření vlhkosti. Vlastní popis komunikace je popsán výše v kapitole věnované snímači SHT11 (viz kap.2.2).

V dalším kroku jsou volány funkce pro přepočet 14-ti bitového čísla na hodnotu teploty ve stupních (`calc_sht11_temp`), převod 12-ti bitového čísla na hodnotu vlhkosti v procentech (`rhcalc_int`) a výpočet teploty rosného bodu. Návrátová hodnota funkce pro výpočet teploty vzduchu a hodnota teploty rosného bodu jsou ve stupních vynásobených deseti. Proto je třeba tyto hodnoty před zobrazením převést.

```
error=s_measure( &teplota_raw,0);
if (error==0){
    error=s_measure( &vlhkost_raw,1);
}
if (error==0){
    temp=calc_sht11_temp(teplota_raw);
    rh=rhcalc_int(vlhkost_raw);
    dew=calc_dewpoint(rh,temp);
}
```

Výpočet nadmořské výšky je prováděn na základě naměřené teploty, atmosférického tlaku a hodnoty tlaku uložené v paměti jako kalibrační konstanta. Tato konstanta udává atmosférický tlak ve výchozím bodě.

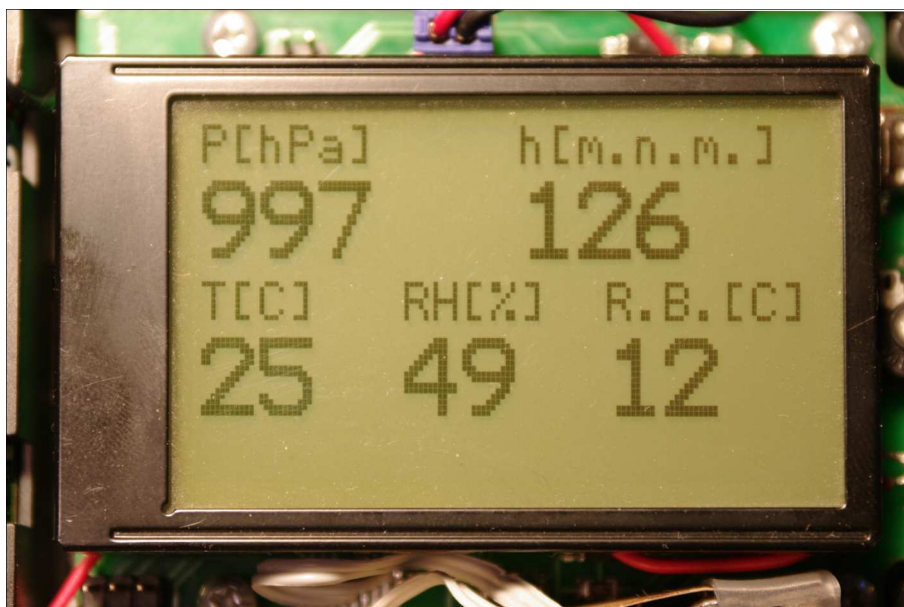
```
vyska=-log(tlak/tlak1)*29.15*(temp/10.0+273.15);
```

5.4 Výpis na displej

Pro zobrazování měřených hodnot na displej jsou použity knihovny `ks0108.c` a `ks0108.h` obsahující funkce pro ovládání displeje a soubory `font6x8.h` a `font12x16.h` obsahující znakové sady. V hlavičkovém souboru `ks0108.h` je mimo jiné nastaven port, na který je připojena datová sběrnice displeje a jednotlivé piny mikrokontroléru, na které jsou připojeny řídicí signály (viz Obr. 4.6 a Tab. 4.7).

Před vlastním použitím knihovny je potřeba si v kódu vytvořit strukturu obsahující font kterým budou vypisovány informace na displej. V mém případě jsem to provedl takto:

```
largeFont.width = FONT12X16_WIDTH;  
largeFont.height = FONT12X16_HEIGHT;  
largeFont.charData = Font12x16;  
  
smallFont.width = FONT6X8_WIDTH;  
smallFont.height = FONT6X8_HEIGHT;  
smallFont.charData = Font6x8;
```



Obrázek 5.2: Zobrazení aktuálních hodnot na displeji

Pokud máme vloženy správné hlavičkové soubory a vytvořenou strukturu s fontem, můžeme přistoupit k vlastnímu vypisování. Knihovna se používá tak, že nejprve inicializujeme displej funkcí `ks0108Init()` poté se nastaví pozice na displeji odkud bude text vypisován a to funkcí `ks0108GotoXY(X,Y)`. Vlastní vypsání textu na displej se provede funkcí `ks0108PutString("retezec", font)`, kde `retezec` je vypisovaný textový řetězec a `font` je struktura obsahující písmo kterým bude text vypsán. V našem případě to jsou písma o velikosti 6x8 nebo 12x16 bodů. Na výpisu níže je ukázka zapsání jednoho řádku s informací o aktuálním atmosferickém tlaku. Protože pro měřené a vypočítané hodnoty jsou použity proměnné typu `float` jsou hodnoty před výpisem formátovány funkcí `sprintf()`.

```

sprintf(tlakhpa, "%.1f hPa", tlak);
ks0108Fill(CLEAR);
ks0108GotoXY(0,4);
ks0108PutString("Tlak:", smallFont);
ks0108GotoXY(56,04);
ks0108PutString(tlakhpa, smallFont);

```

5.5 Sériový přenos dat

Pro sériové odeslání dat do počítače využívám rozhraní USART (Universal Synchronous Asynchronous Receiver Transmitter) procesoru a převodník FTDI FT232R pro převod na rozhraní USB. Mikrokontrolér má dvě sériové rozhraní USART, já jsem použil první, takže všechny registry použité pro komunikaci jsou označeny nulou (UCSROA, UDR0, RXEN0,..) pokud bychom chtěli použít druhý kanál, místo nul by byly použity jedničky. Význam jednotlivých registrů je popsán výše v popisu použitého mikrokontroléru.

Odeslání dat do PC je prováděno funkcí `USART_Print`. Tato funkce ukládá postupně do registru UDR0 znaky které chceme odeslat na sériový kanál. Po úspěšném odeslání je nastaven příznak TXC v UCSRA.

```

void USART_Print (char *string)
{
    while (*string) {
        loop_until_bit_is_set(UCSROA, UDRE0);
        UDR0 = *string;
        string++;
    }
    return;
}

```

Před odesláním řetězců na rozhraní USART je třeba nejprve nastavit parametry komunikace. To provádím pomocí funkce `USART_Init`. Vzhledem k použitému vnitřnímu oscilátoru na kmitočtu 8MHz jsem zvolil přenosovou rychlost 9600 Bd. Při těchto parametrech je maximální procentuální chybovost přenosu 0,2%, což je dostačující. Formát dat je 8 datových bitů a jeden stop-bit.

```

void USART_Init( unsigned int baud )
// Inicializace seriového rozhraní

```

```

{
// nastaveni baudrate
UBRR0H = (unsigned char)(baud>>8);
UBRR0L = (unsigned char)baud;

// povoli RX a TX
UCSR0B = (1<<RXEN0)|(1<<TXEN0);
// RXEN - povoli příjem
// TXEN - povoli vysílání

// Nastavení formátu dat: 8N1
UCSR0C = (0<<USBS0)|(3<<UCSZ00);
// USBS - počet stopbitů (0=1, 1=2 stopbity)
// UCSZ - délka znaku
}

```

Obdobně jako funkce `USART_Print` odešle data na sériový kanál, další použitá funkce `USART_Read` ze sériového kanálu data čte. Princip je velmi podobný funkci pro zápis. Nejprve přečteme znak z registru `UDR`, poté kontrolujeme příznak `RXC` v `UCSRA0`.

5.6 Ukládání dat

Pro sběr naměřených dat jsem použil vnitřní EEPROM paměť mikroprocesoru. Velikost této paměti je 4kB. Zápis do paměti je řízen přes následující I/O registry:

- **EEARLH** a **EEARL** - pro uložení, nebo čtení adresy
- **EEDR** - datový reistr - do registru musí být zapsáno před spuštěním zápisu
- **EECR** - řídicí registr - význan jednotlivých bitů je:
 - **EEMWE** - 1 povoluje zápis (p 4. cyklech se shodí do nuly), 0 zakáže zápis
 - **EEWE** - 1 spouští zápis, pokud `EEMWE=1`, 0 ukončení zápisu
 - **EERE** - 1 spustí čtení, 0 konec čtení

Výrobce doporučuje pro čtení dat z EEPROM následující postup. Nejprve počkat až bude `EEWE=0`, poté do `EEARL` a `EEARH` zapsat adresu EEPROM. Dále je potřeba uložit zapisovaná data do `EEDR` a nastavit bit `EEMWE=1`. Během čtyř hodinových cyklů je třeba nastavit `EEWE=1`.

Pro zápis do EEPROM používám funkce `EEPROM_write` a funkci `EEPROM_read`, jejich princip je jednoduchý a vychází z předchozích řádků.

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
    // ceka na dokonce predchoziho zapisu
    while(EECR & (1<<EEMWE))
    ;
    // nastaveni adresy a datoveho registru
    EEAR = uiAddress;
    EEDR = ucData;
    // nahozeni EEMWE do log. 1
    SETB(EECR,EEMWE);
    //EECR |= (1<<EEMWE);
    // spusteni vlastniho zapisu do EEPROM
    SETB(EECR,EEWE);
    //EECR |= (1<<EEWE);
}

unsigned char EEPROM_read(unsigned int uiAddress)
{
    // ceka na dokonce predchoziho zapisu
    while(EECR & (1<<EEMWE))
    ;
    // nastaveni adresy
    EEAR = uiAddress;
    // zacatek zapisu
    SETB(EECR,EERE);
    // navratova hodnota - data z registru
    return EEDR;
}
```

První dvě hodnoty v paměti jsou obsazeny pomocnými proměnnými. První obsahuje kalibrační konstantu pro výchozí tlak a druhá hodnota je adresa v paměti na které je uložena poslední hodnota. Dále pměť EEPROM obsahuje vlastní naměřená data. Zápis je prováděn postupně až do doby, kdy je paměť zaplněna.

6 VÝSLEDKY PRÁCE

6.1 Osazení a realizace hardwarové části

Výsledkem mé práce je funkční vzorek přístroje. S vlastním osazením a oživením přístroje nebyly větší problémy. Nejproblematictější částí byly napájecí obvody. Zapojení napájecích DC/DC měničů je v podstatě totožné s zapojením v katalogovém listu obvodu. Proudové zatížení nepřekračuje katalogové hodnoty. Bohužel i tak se ukázalo, že měniče jsou poněkud nestabilní. Při napájení ze dvou 1,5V baterií se obvod choval značně nestabilně. Po přidání třetí baterie se situace zlepšila a vše funguje jak má. Vzhledem k rozměrům krabičky a požadavkům na napájení by bylo možné napájení vyřešit použitím 9V baterie, která by přímo mohla napájet zdroj referenčního napětí (REF02CS) a lineární pevný regulátor například 7805 pro napájení zbylých obvodů. Tato úprava by snížila náklady na součástky a odpadlo by zdržení způsobené hledáním závady při napájení z dvou 1,5V baterií.



Obrázek 6.1: Výsledné mechanické provedení zařízení

6.2 Software a test funkčnosti

Po osazení přístroje jsem jako první zprovoznil knihovny pro výpis na displej. Pak už byla práce na vlastním řídicím softwaru poměrně snadná. Díky výpisu pomocných proměnných na displej se už dal software dobře ladit. Software nahraný v testovacím prototypu splňuje všechny požadavky zadání. Po zapnutí přístroje je možné pomocí

jenoduchého menu zapnout záznam do paměti, provést kalibraci, či odeslat data po USB. Po úplném nastartování zařízení vypisuje všechny požadované hodnoty. Zobrazována je teplota, tlak, nadmořská výška, relativní vlhkost vzduchu a rosný bod.

6.3 Parametry přístroje

Vzhledem k použitému 12b převodníku a citlivosti čidla (45,9 mV/kPa) vychází výsledná rozlišovací schopnost na 0,26 hPa. S použitím barometrické rovnice tedy zjistíme, že rozlišení výškoměru jsou 2 m. Jelikož uvažujeme výsledné použití pro turistické účely zdála se tato hodnota jako rozumná. Naměřený tlak jsem porovnával s komerčně vyráběnou meteostanicí (Hama EWS-900) a hodnoty naměřeného tlaku byly shodné. Pokud bycho požadovali větší rozlišení a přesnější výsledky, bylo by třeba použít přesnější A/D převodník. Například s 16b převodníkem bychom se dostali již na rozlišení 15 cm. Přesnějších hodnot bez nutnosti kalibrace by bylo samozřejmě možné dosáhnout například použitím technologie GPS. To by však zdražilo celý přístroj, pokud by byl GPS modul přidán navíc. Jelikož nás zajímají také meteorologické údaje, bylo by škoda GPS modulem nahradit například snímač tlaku.

Rozlišovací schopnost snímače SHT11 je pro měření teploty 0,01 C při 14-ti bitovém měření. Pro měření vlhkosti je pak rozlišovací schopnost 0,03 % pro 12-ti bitové měření.

Tabulka 6.1: Parametry přístroje

Napájení	
Napájecí napětí	2,7 - 4,5 V
Spotřeba proudu	40 - 90 mA
Zdroj napájení	3x AAA baterie
Rozsah měřených hodnot	
Tlak	150 - 1150 hPa
Teplota	-40 - 123,8 C
Vlhkost relativní	0 - 100 %
Rozměry a hmotnost	
Šířka	135 mm
Výška	70 mm
Hloubka	24 mm
Hmotnost	160 g

7 ZÁVĚR

Závěrem bych zhodnotil výsledky a průběh své práce. Nejprve jsem prostudoval základy měření meteorologických veličin a seznámil se s dostupnými senzory. Vybral jsem snímač tlaku Motorola MPX4115 a Sensirion SHT11 pro snímání vlhkosti. Následovala volba vhodného procesoru, ta padla na mikrokontrolér Atmel AVR ATmega64, jenž vyhovuje jak dostatečnou pamětí, tak počtem vstupně výstupních portů. V další fázi jsem navrhl schéma zapojení a tištěný spoj jehož rozměry byly voleny s ohledem na velikost vybrané krabičky. Po zhotovení tištěného spoje následovalo osazení, oživení a otestování základní funkčnosti.

Další částí práce byl návrh vlastního softwaru, který zařízení řídí. Důležité bylo napst základní funkce, která program opakovaně používá. Těmito jsou například funkce pro čtení dat ze snímače a A/D převodníku, zápis a čtení z EEPROM paměti, přenos dat po rozhraní USART a výpis na displej. Na závěr jsem na základě těchto funkcí napsal program, který zařízení řídí.

REFERENCE

- [1] ATMEL. *AVR 910: In-System Programming* [online]. Dostupné z URL: <<http://www.atmel.com/atmel/acrobat/doc0943.pdf>>.
- [2] ATMEL. *Serial Peripheral Interface Summary* [online]. Dostupné z URL: <<http://www.atmel.com/atmel/acrobat/6088S.pdf>>.
- [3] ATMEL. *8-bit Microcontroller with 64K Bytes In-System Programmable Flash* [online]. Dostupné z URL: <<http://www.atmel.com/atmel/acrobat/doc2490.pdf>>.
- [4] MOTOROLA. *MPX4115A, MPXA4115A SERIES* [online]. 2001. Dostupné z URL: <http://www.freescale.com/files/sensors/doc/data_sheet/MPX4115A.pdf>.
- [5] SENSIRON. *SHT1x/SHT7x Humidity Sensor* [online]. 2006. Dostupné z URL: <http://www.sensirion.com/en/pdf/product_information/Data_Sheet_humidity_sensor_SHT1x_SHT7x_E.pdf>.
- [6] ELECTONIC ASSEMBLY. *LCD-GRAFIK MODUL 128x64 PIXEL* [online]. 2004. Dostupné z URL: <<http://www.qscomp.cz/Pdf/dip128-6.pdf>>.
- [7] COMAPQ, HEWLETT-PACKARD, INTEL, LUCENT, MICROSOFT, NEC, PHILIPS. *Universal Serial Bus Specification* [online]. 2000. Dostupné z URL: <http://www.usb.org/developers/docs/usb_20_05122006.zip>.
- [8] SKŘEHOT, P. *Úvod do studia meteorologie*. Praha: Meteorologická Operativní Rada, 2004.
- [9] HW SERVER *RS232* [online]. 2003. Dostupné z URL: <<http://rs232.hw.cz>>.
- [10] WIKIPEDIE-OTEVŘENÁ ENCYKLOPEDIE *Barometrické měření výšky* [online]. 2008. Dostupné z URL: <http://cs.wikipedia.org/wiki/Barometrické_měření_výšky>.
- [11] MICROCHIP. *MCP3202 12-bit Analog-to-Digital Converter* [online]. 2001. Dostupné z URL: <<http://ww1.microchip.com/downloads/en/DeviceDoc/21034D.pdf>>.
- [12] FTDI. *FT232R USB UART I.C.* [online]. 2008. Dostupné z URL: <http://www.ftdichip.com/Documents/DataSheets/DS_FT232R.pdf>.
- [13] METOFFICE. *Met Office: Weather and climate change* [online]. 2008. Dostupné z URL: <<http://www.metoffice.gov.uk/>>.

- [14] BOSS Enclosures. *BOSS Enclosures — IR Pocket Cases, Low-profile Control Enclosures, UL94 V0 ABS Enclosures* [online]. 2009. Dostupné z URL: <<http://www.boss-enclosures.co.uk/>>.

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

A/D Analogově digitální převodník

ALU Aritmeticko logická jednotka - Arithmetic Logic Unit

ATM Atmosféra

DMA Přímý přístup do paměti - Direct Memory Access

GPR Registr pro všeobecné použití - General Purposes Register

IRQ Požadavek přerušení - Interrupt Request

ISP Programování přímo na desce - In System Programming

JTAG Standard pro testování plošných spojů, pamětí a dalších obvodů - Joint Test Action Group

LUT Vyhledávací tabulka - Look Up Table

MCU Mikrokontrolér - Microcontroller Unit

MIPS Počet miliónů instrukcí za vteřinu - Millions of Instructions Per Second

PWM Pulsně šířková modulace - Pulse Width Modulation

RISC Počítač s redukovanou instrukční sadou - Reduced Instruction Set Computer

SPI Sériové periferní rozhraní - Serial Periperial Interface

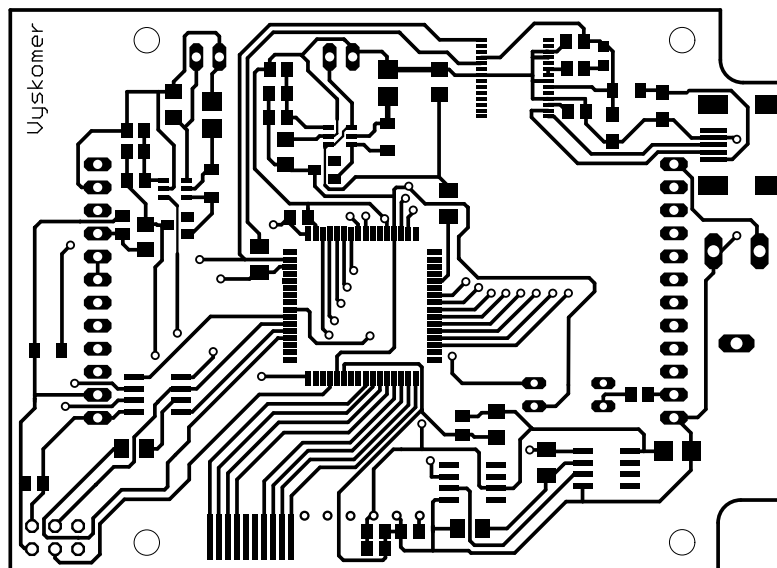
TTL Tranzistorově-tranzistorová logika - Transistor-Transistor Logic

USB Univerzální sériové rozhraní - Universal Serial Bus

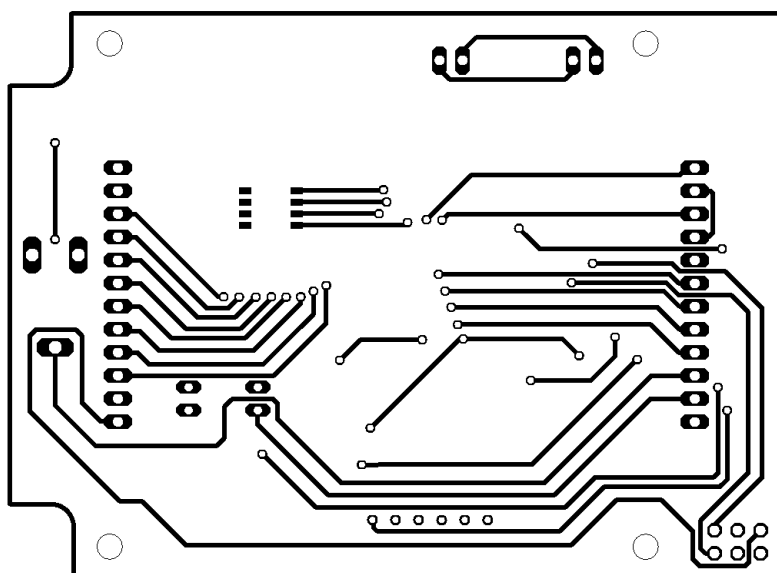
SEZNAM PŘÍLOH

A	Výkresová dokumentace	52
A.1	Schéma zapojení	52
A.2	Tištěný spoj - strana spojů	53
A.3	Tištěný spoj - strana součástek	53
A.4	Osazovací plánec - vrchní strana	54
A.5	Osazovací plánec - spodní strana	54
A.6	Tištěný spoj - deska s tlačítky	55
A.7	Přístrojová krabička	56
B	Seznam součástek	57
C	Software - vývojový diagram	59

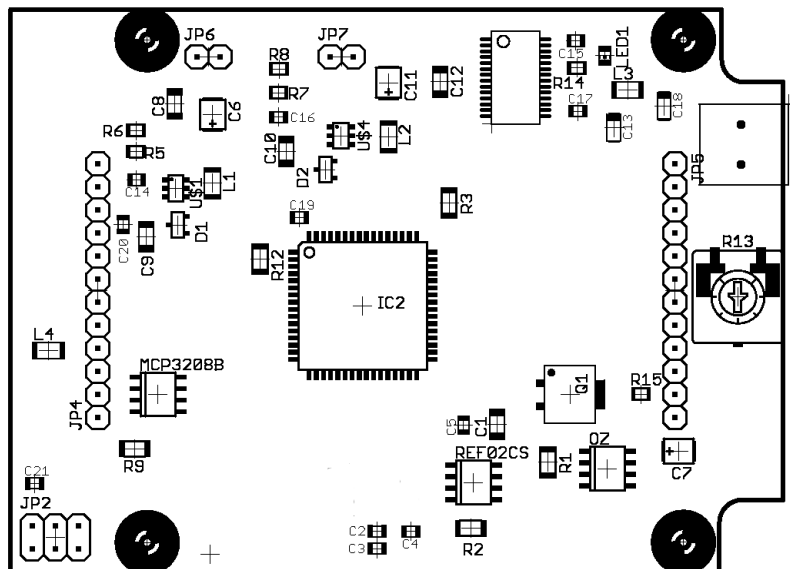
A.2 Tištěný spoj - strana spojů



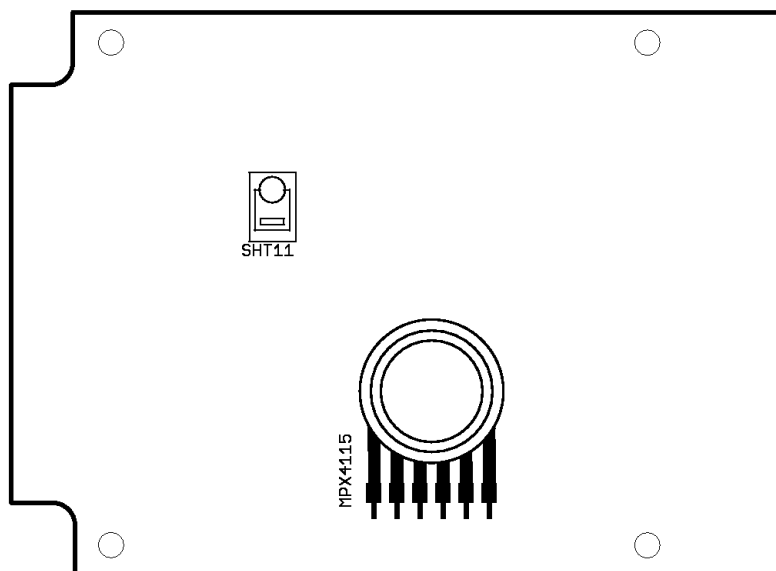
A.3 Tištěný spoj - strana součástek



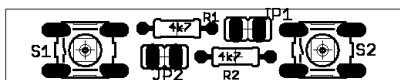
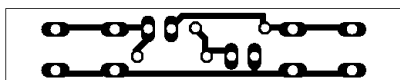
A.4 Osazovací pláněk - vrchní strana



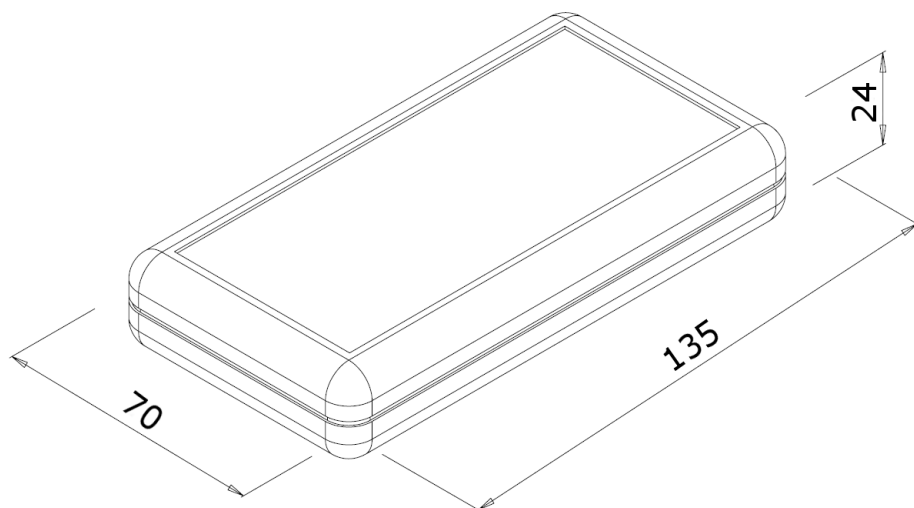
A.5 Osazovací pláněk - spodní strana



A.6 Tištěný spoj - deska s tlačítky



A.7 Přístrojová krabička



B SEZNAM SOUČÁSTEK

C1, C5, C15, C17, C19	100n
C2, C18	10n
C3	1u
C4	470p
C6	4u7
C7	
C8, C9, C10	1u, X7R
C11, C13	4u7
C12	1u, X7R
C14, C16	10p
D1	BAT54SMD
D2	BAT54SMD
IC2	ATMEGA64-A
JP2	PINHD-2X3
JP4, JP5	PINHD-1X12
JP6, JP7	PINHD-1X2
L1, L2, L3, L4	
LED1	SMD LED
LED2, LED3	3MM
MCP3208B	ADC08032D
OZ	LM1458D
R1	69k8
R2, R3	10k
R4, R10	5k
R5, R7	2M2
R6, R8	390k
R9, R12	0R
R11	270R
REF02CS	REF02D
SHT11	SENSIRION SHT11

U1 MAX8570
U2 FT232R
U3 USBMINIB
U4 MAX8570
U5 Motorola MPX4115

C SOFTWARE - VÝVOJOVÝ DIAGRAM

